Organização de programas em Python

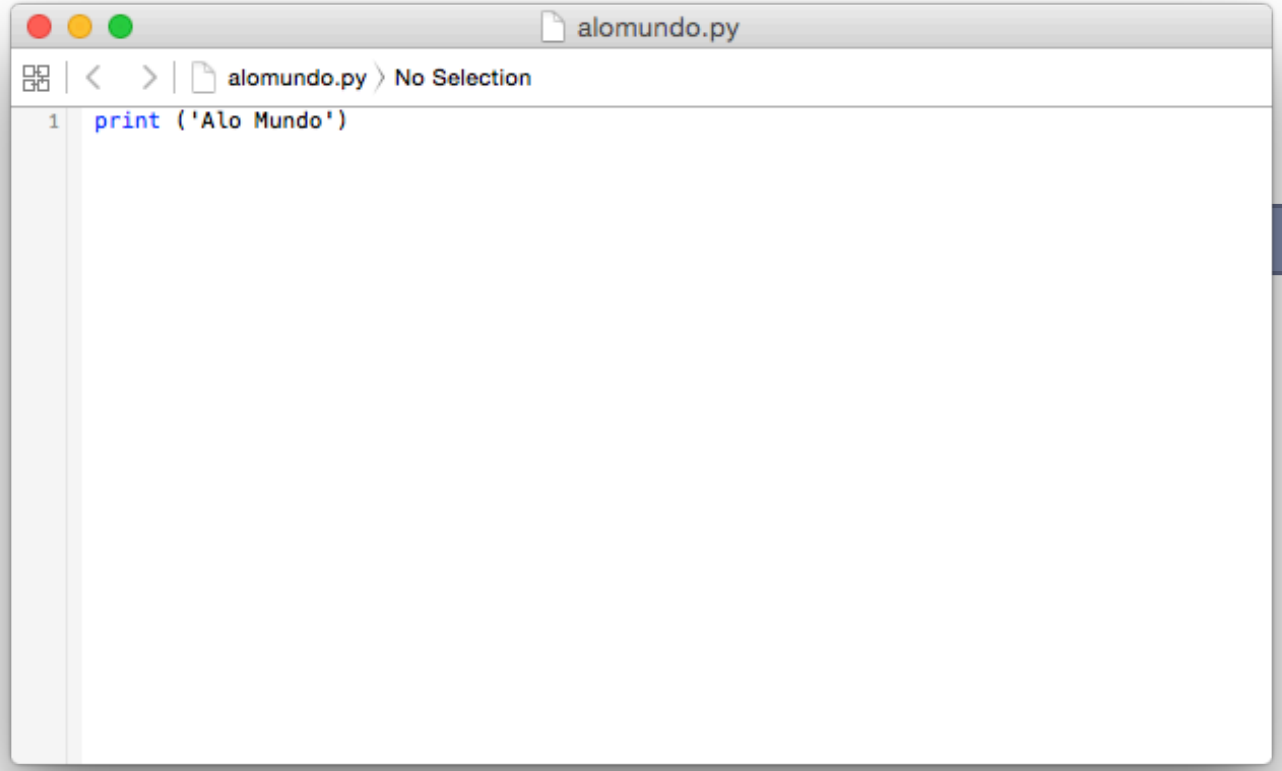
Loana T. Nogueira
loanatito@id.uff.br

Vamos programar em Python! Mas...

- Como um programa é organizado?
- Quais são os tipos de dados disponíveis?
- Como variáveis podem ser declaradas?
- Como atribuir valores às variáveis?
- Como entrada e saída básica de dados podem ser feitas?

Vamos começar com um exemplo...

Primeiro passo: escrever o programa!



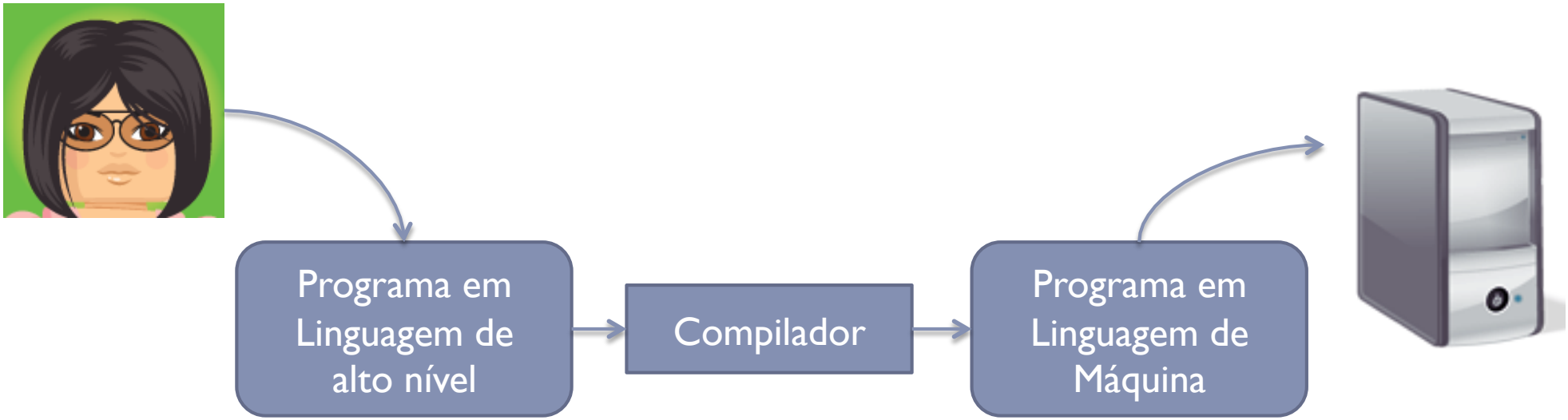
alomundo.py

Mas o computador não conhece Python!!!

- O computador só entende binário
  - Linguagem de zeros e uns
  - 0100100111010101010010010101, entendeu?
- Precisamos traduzir o programa Python para binário

Compilação

- Na maioria das linguagens, antes de executar um programa, é necessário compilar o programa
- O compilador gera um arquivo "executável"
  - Esse novo arquivo é o que será de fato executado



Python é uma linguagem interpretada

- Não é necessário compilar o código Python
- O interpretador Python vai lendo o código fonte, traduzindo para linguagem de máquina e executando ao mesmo tempo

Instalação do Interpretador Python

- Download do Python 3.8.6
  - http://www.python.org/downloads/

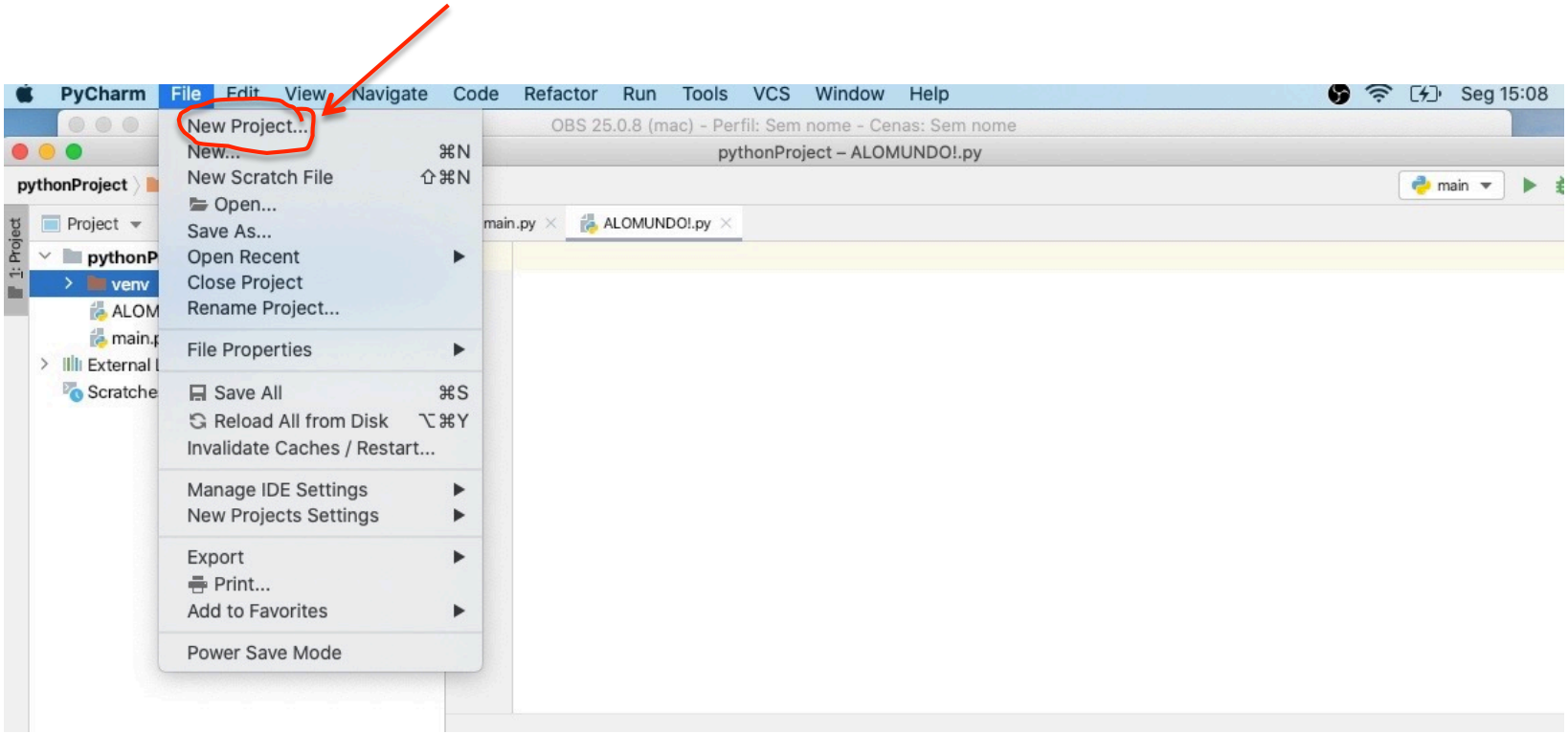
Notepad x IDE

- Dificuldades do Notepad
  - Editor básico, sem ajuda para programar
  - Execução externa
- *Integrated Development Environment* (IDE)

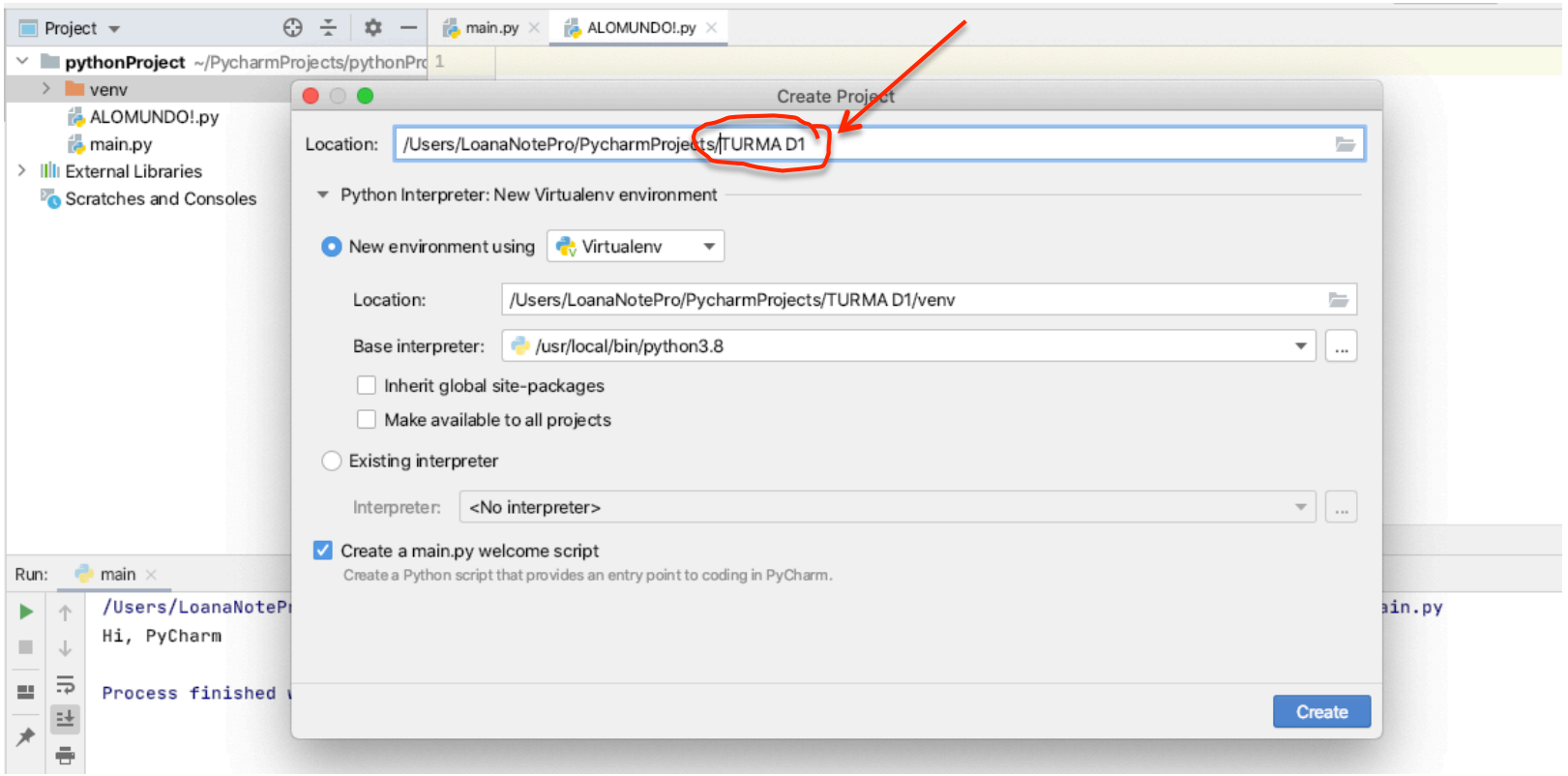
Instalação do PyCharm

- Usaremos o PyCharm nas aulas, mas os alunos podem optar por qualquer outra IDE ou editor
- Download do PyCharm
  - http://www.jetbrains.com/pycharm-educational/

Criando um projeto no PyCharm...

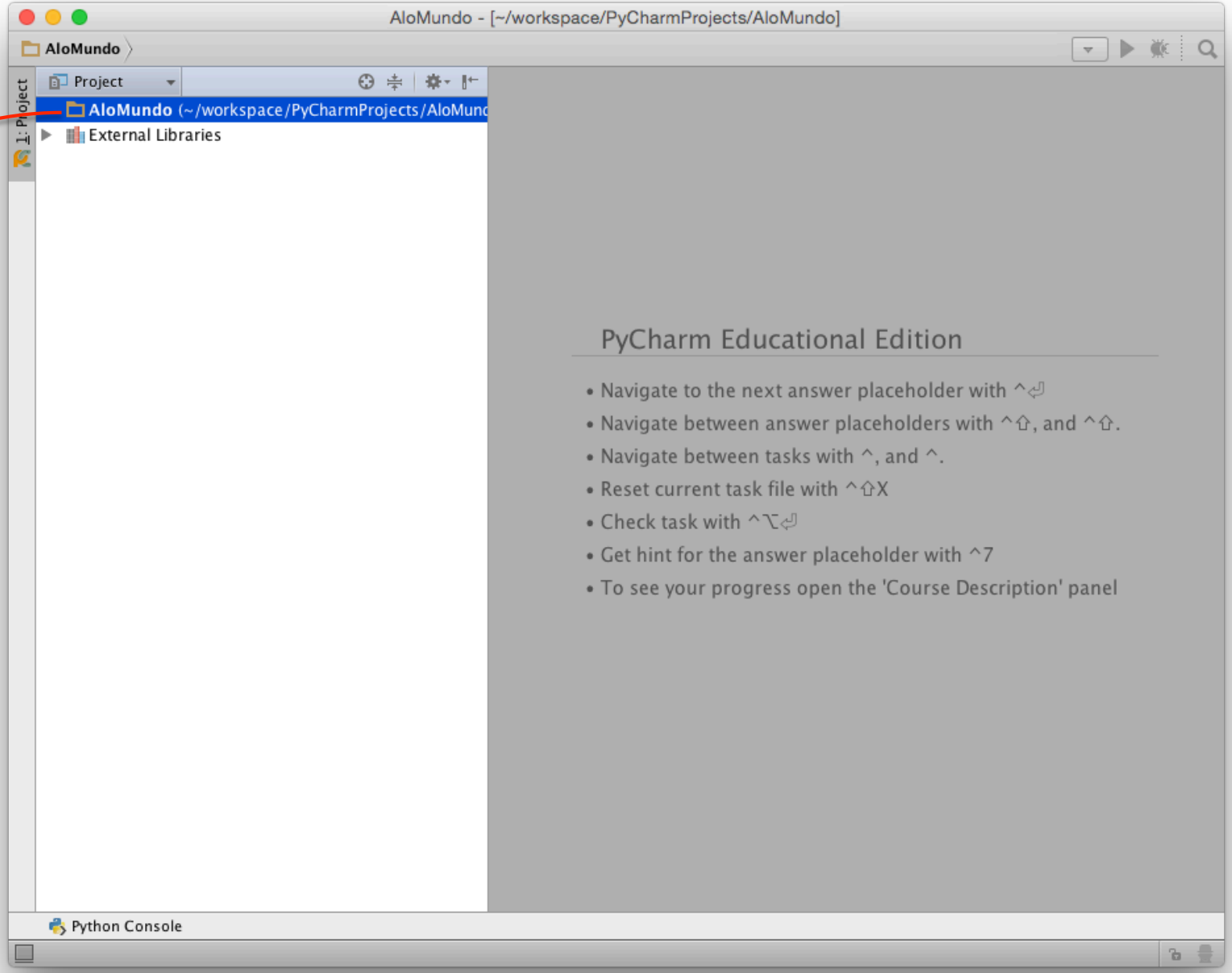


Criando um projeto no PyCharm...



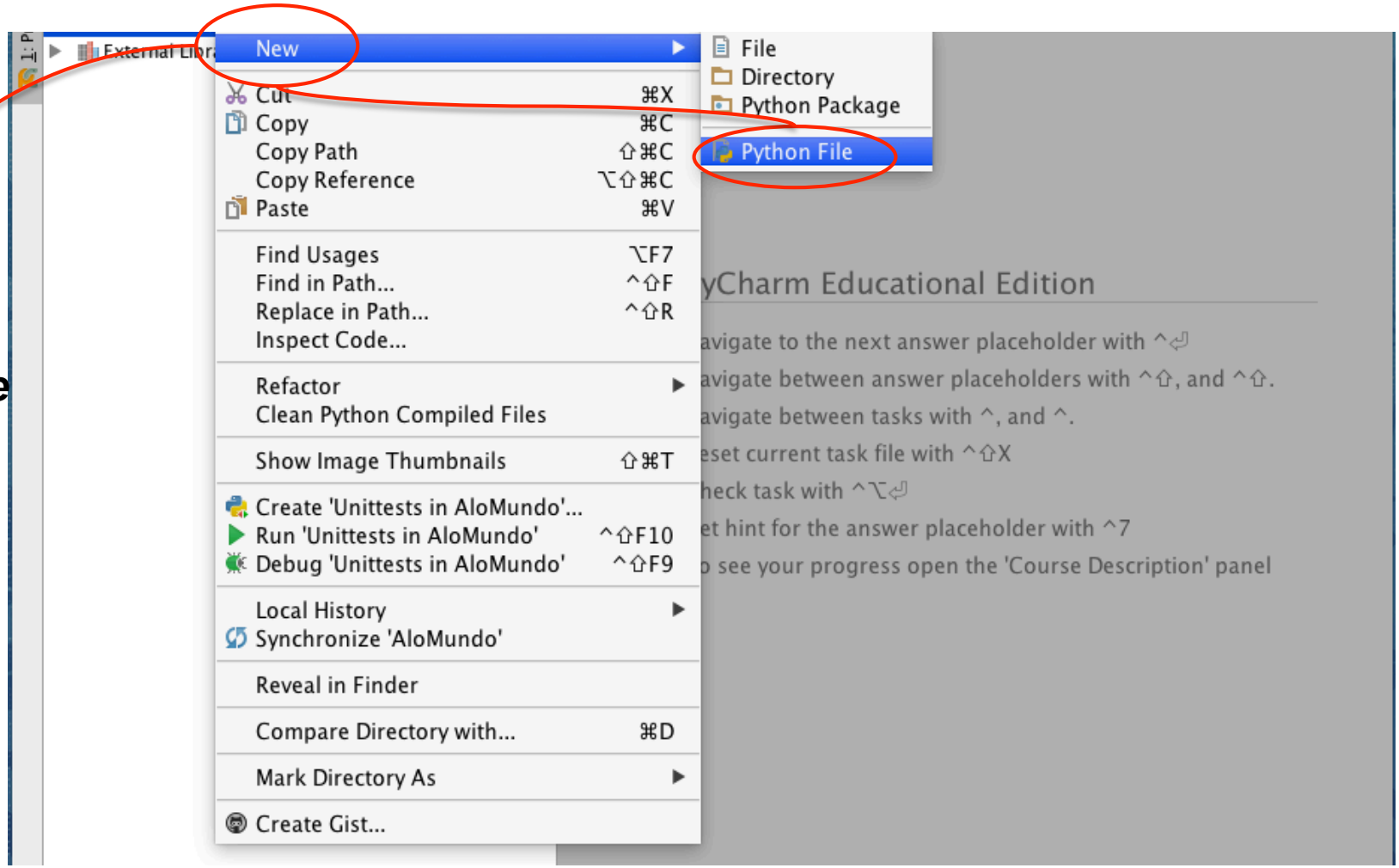
Criando um Arquivo Python no Projeto

Clicar com o **botão da direita** sobre o nome do projeto



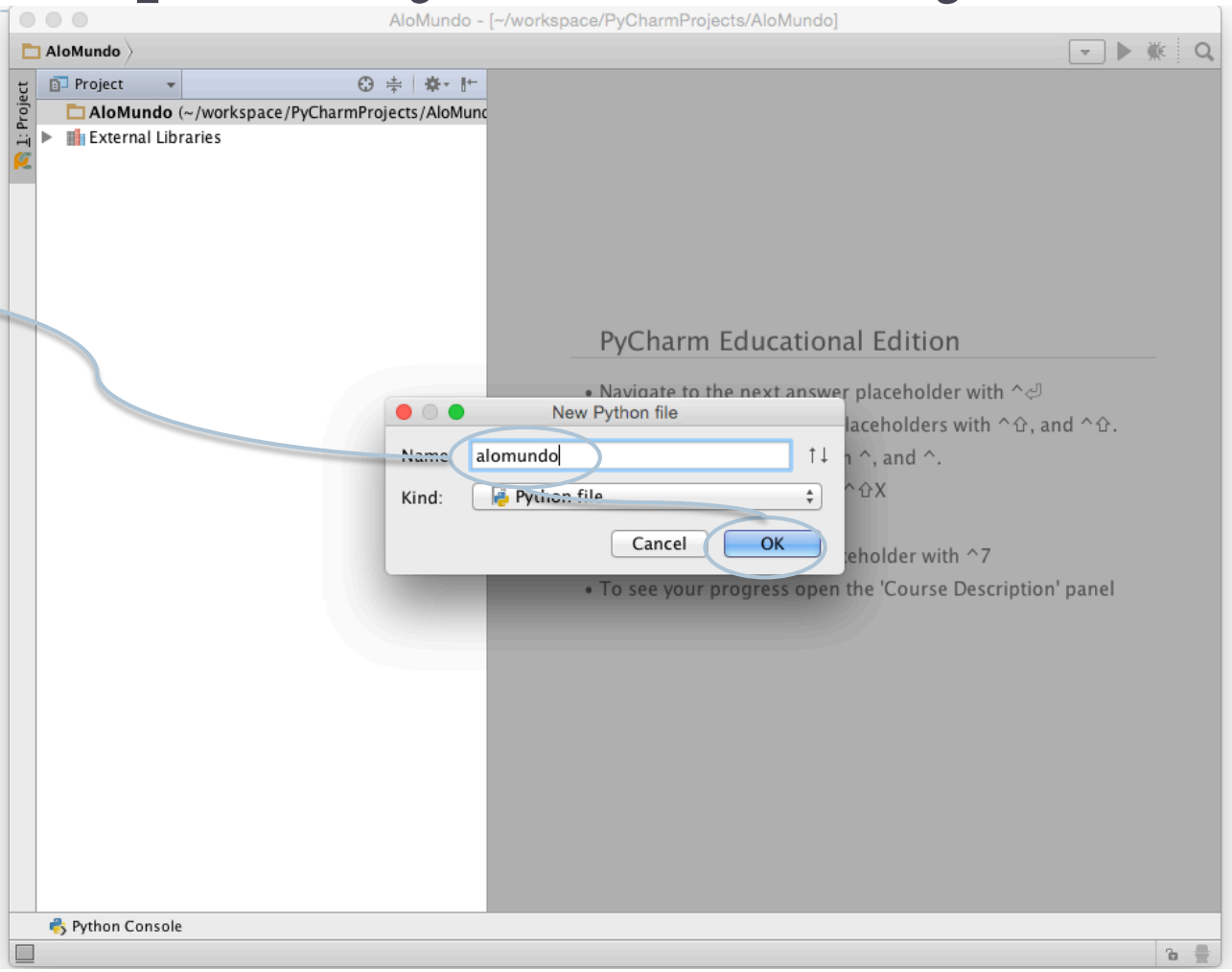
Criando um Arquivo Python no Projeto

Selecionar ***New*** / ***Python File***

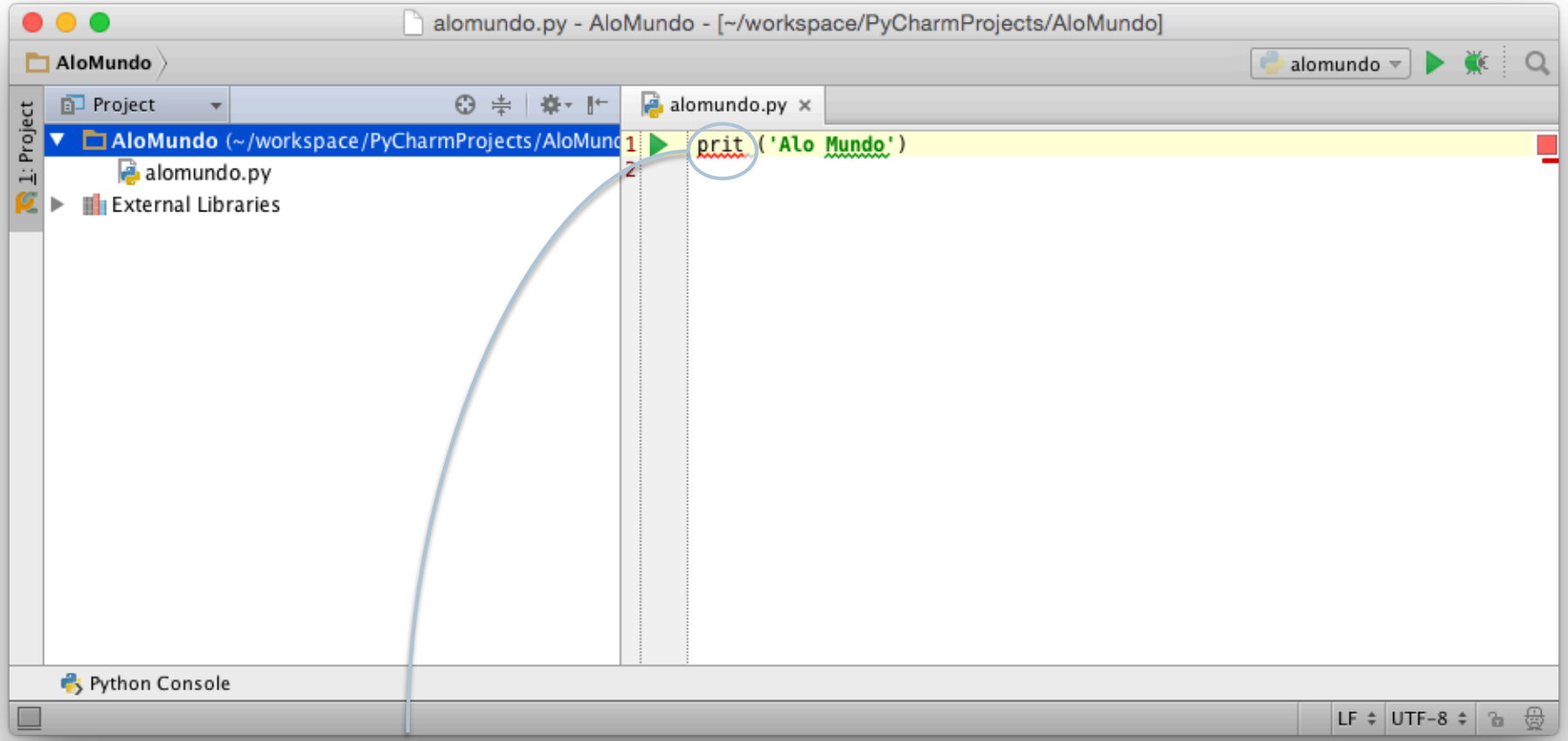


Criando um Arquivo Python no Projeto

Informar o **nome** do arquivo e depois clicar em ***OK***



Escrevendo o programa no PyCharm...

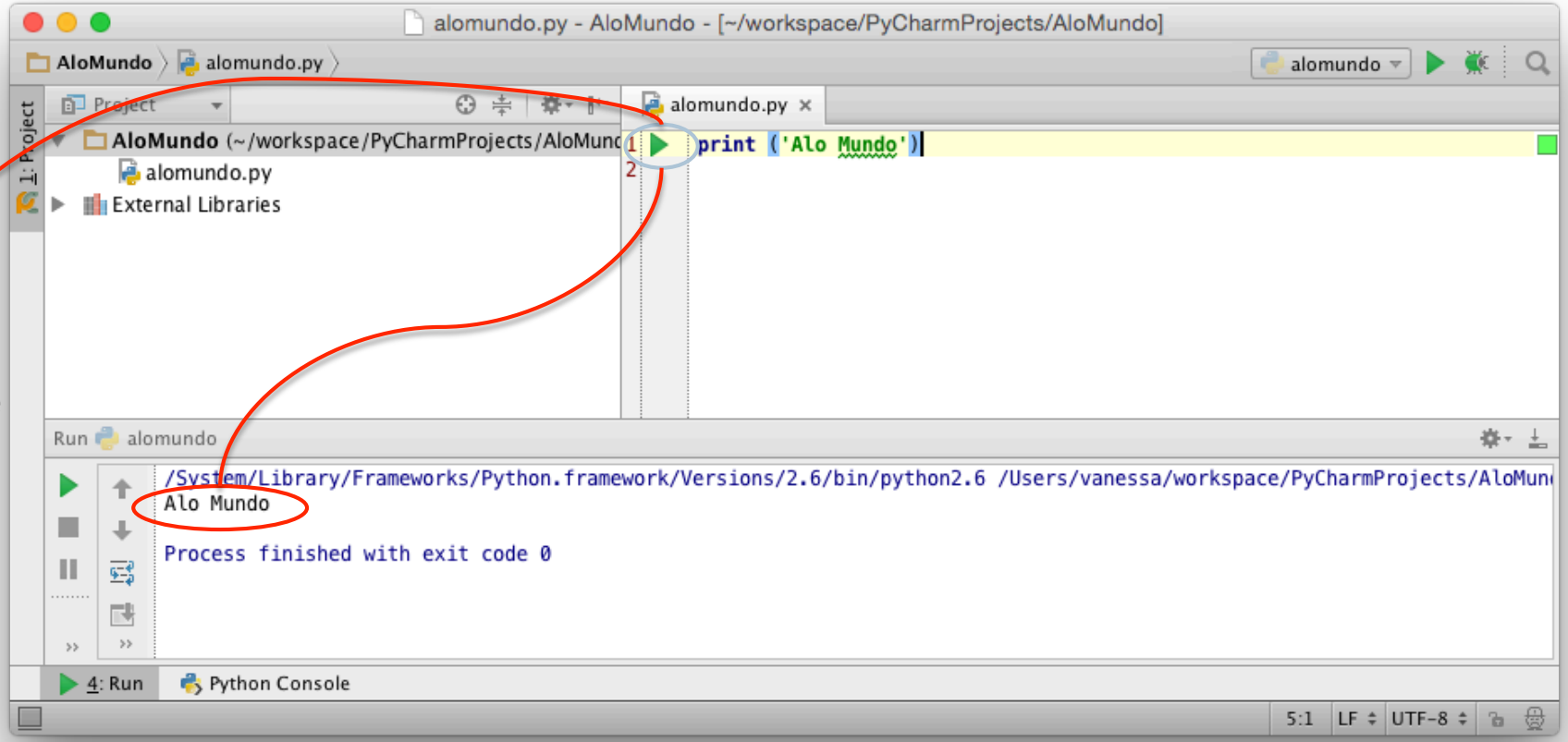


Avisos sobre erros durante a edição do código

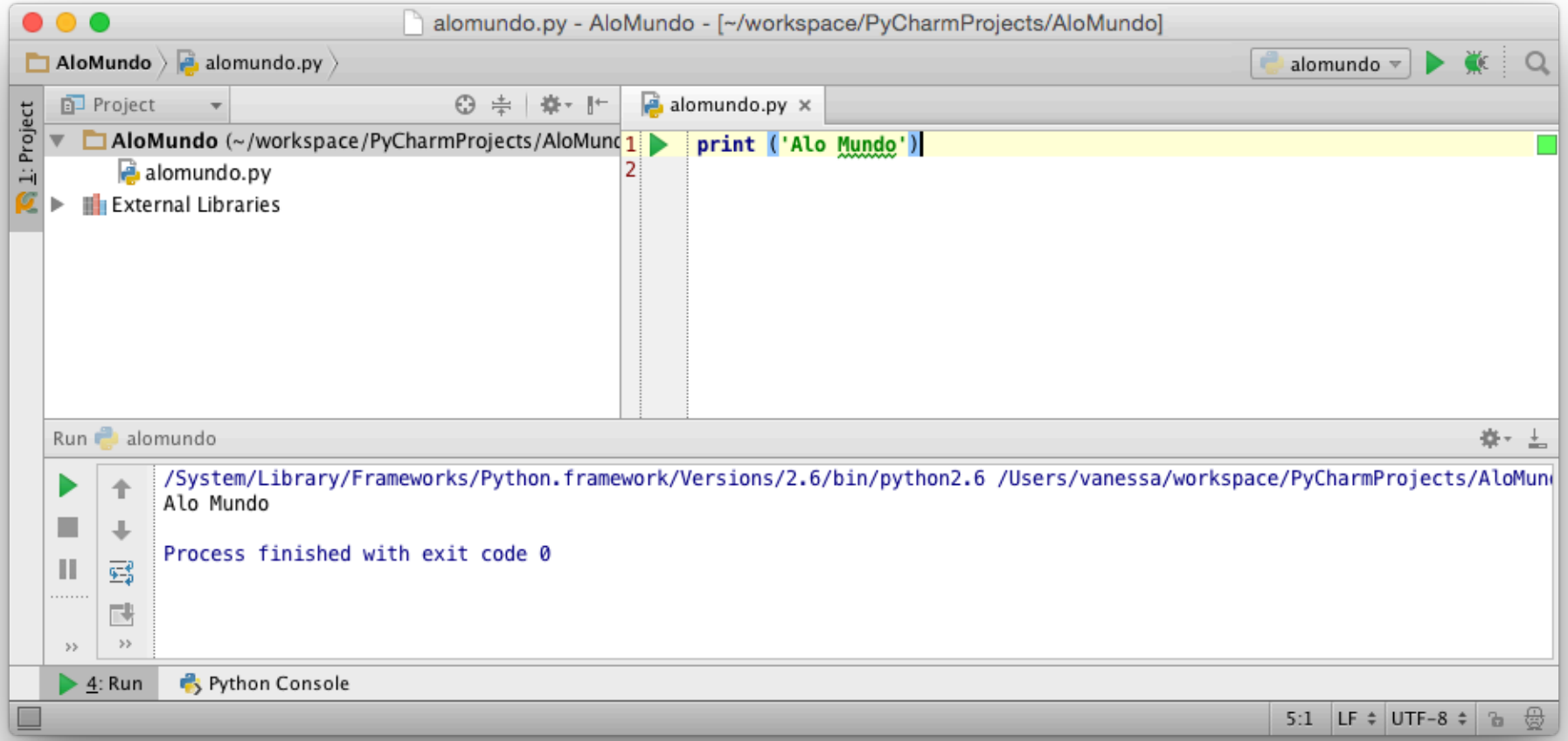
Executando o programa no PyCharm...

Clicar neste ícone para **executar o programa**

No painel inferior ocorrerá a **entrada e saída de dados**



Escrevendo e executando o programa no PyCharm...



VAMOS FAZER JUNTOS?

Regras básicas

- Sequência dos comandos é importante
- Blocos devem ser criados usando indentação (com espaços ou tab)

Comentários

- Comentários são trechos do programa voltados para a leitura por humanos, e ignorados pelo interpretador
- Começam com o símbolo **#**
  - Tudo na linha após **#** é ignorado pelo interpretador

- Use comentários para documentar seu código e fazer com que ele seja fácil de entender por outras pessoas

Atribuição de valores

- Em Python, o operador de igualdade (=) é usado para atribuir valores às variáveis
- É equivalente ao símbolo de atribuição (←) que usávamos no pseudocódigo
- Sempre na forma: **variável = valor ou expressão**
  - A expressão do lado direito é processada
  - O valor gerado é atribuído à variável

Exemplo de programa em Python

```
# Este programa calcula a area de um triangulo retangulo
altura = 15
base = 3
area = (altura * base)/2
print(area)
```

Quais são os tipos de dados disponíveis?

- Em Python, toda variável tem um tipo
- Com isso, o computador pode saber quais operações são permitidas
- Os tipos podem ser divididos em três grupos
  - Tipos numéricos (inteiro, float, ...)
  - Tipos textuais (caractere e string)
  - Tipo lógico (booleano)
- Os tipos são definidos dinamicamente, pelo próprio Python
  - Não é preciso dizer de que tipo é cada variável

Exemplo de variáveis lógicas (**boolean**)

```
x = True
y = False
```

Exemplo de variáveis textuais (**string**)

```
nome = 'Maria'
sobrenome = "Silva"
letra = 'A'
texto = 'Alo Mundo'
```

Exemplos de variáveis numéricas

```
a = -5
b = 10
c = 200
d = -12312312
e = 345092834
f = 2.5
g = 0.6023e24
h = 0.4e-3
```

Tipagem Dinâmica

```
a = -5          → inteiro
b = 10          → inteiro
c = 200         → inteiro
d = -12312312   → inteiro
e = 345092834   → inteiro
f = 2.5         → float
g = 0.6023e24   → float
h = 0.4e-3      → float
```

- Tipo é determinado **automaticamente** pelo Python no momento de criação da variável

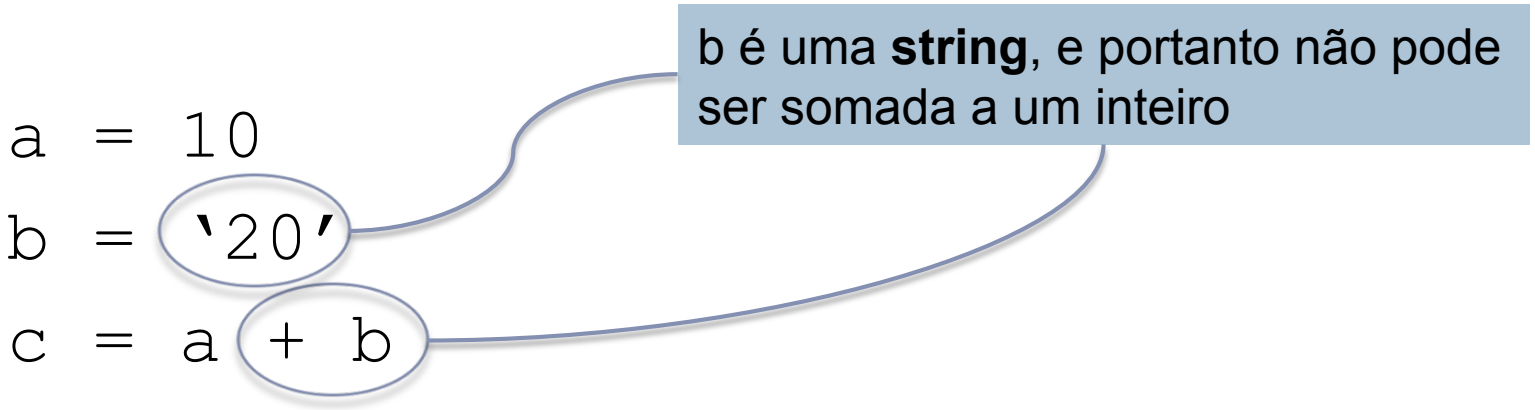
Tipagem Forte

- Uma vez que uma variável tenha um valor de um tipo, ele não pode ser usado como se fosse de outro tipo
- Exemplo:

```
a = 10
b = '20'
c = a + b
```

Tipagem Forte

- Uma vez que uma variável tenha um valor de um tipo, ele não pode ser usado como se fosse de outro tipo
- Exemplo:



Regras para nomes de variáveis

- Os nomes de variáveis devem respeitar algumas regras
  - São sensíveis a caixa
  - Podem ter tamanho ilimitado (mas evite abusos)
  - Devem começar com letra ou underline (_)
  - Outros caracteres podem ser letras, números ou underline
  - Não podem ter espaço nem acentos
  - Não podem ser uma palavra reservada da linguagem

Entrada de dados

- Para entrada de dados, usamos **input**
- É possível informar um texto que aparecerá impresso na tela para que o usuário saiba que o programa está esperando a entrada de um valor

```
nome = input('Digite o nome do aluno: ')
print(nome)
```

Input lê dados como string

- Você pode usar o comando type para saber o tipo que o Python atribuiu a uma variável

```
altura = input('Digite a altura do triangulo: ')
print(type(altura))
base = input('Digite a base do triangulo: ')
print(type(base))
...
```

Mudança de tipo

- Usar int(), float() ou eval() para fazer o Python ler variáveis de tipo numérico

```
altura = int(input('Digite a altura do triangulo: '))
print(type(altura))
base = int(input('Digite a base do triangulo: '))
print(type(base))
area = (base * altura)/2
print('A area do triangulo eh: ', area)
```


Importante: isso só funciona a partir do Python 3.4.3

Saída de dados

- Para saída de dados, usamos **print**

Exemplo de entrada e saída de dados

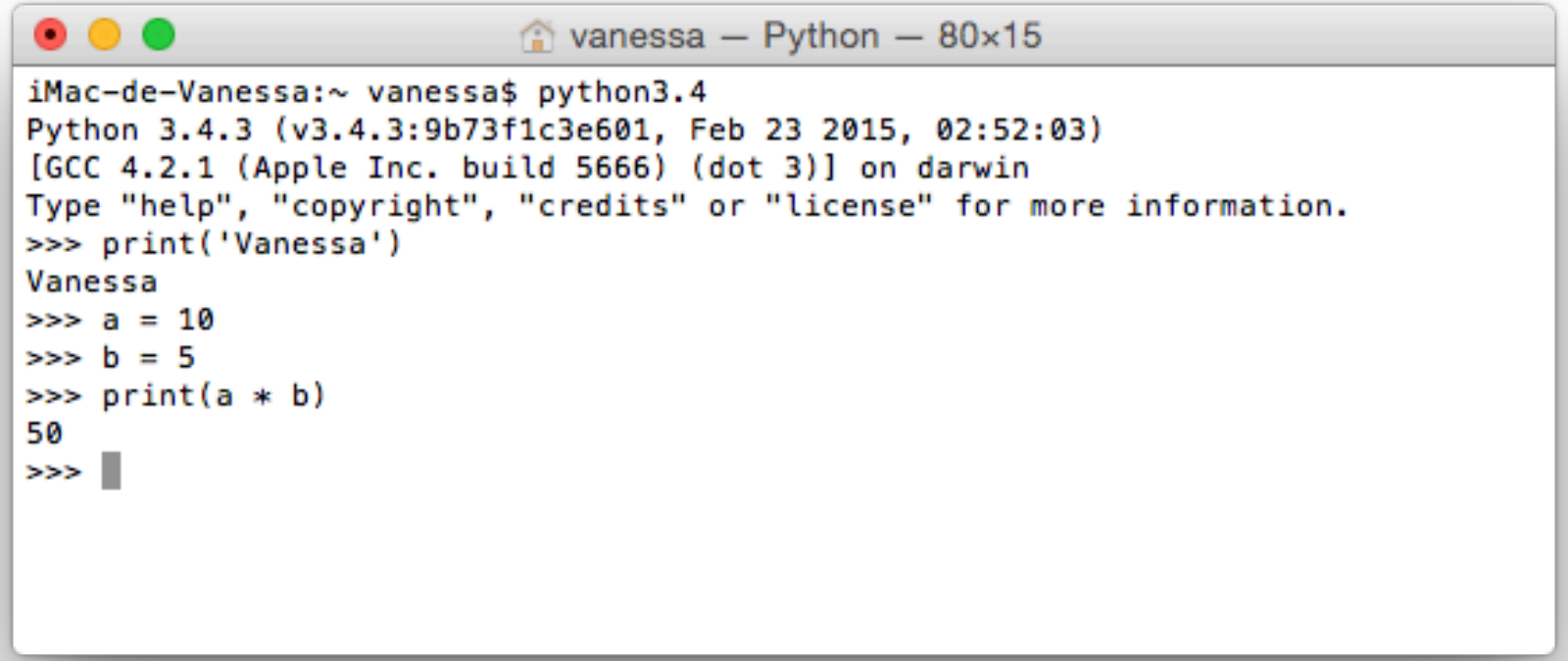
```
print('Prog I é muito legal')
print(123)
altura = 10
print(altura)
print('Vamos pular uma linha \n')
print('O nome do aluno eh', nome)
```

Voltando ao exemplo de programa em Python

```
altura = int(input('Digite a altura do triangulo: '))
base = int(input('Digite a base do triangulo: '))
area = (base * altura)/2
print('A area do triangulo eh:', area)
```

IDLE

- Python também fornece uma interface interativa para execução de pequenas sequencias de comandos
- Basta chamar python no prompt



Exercícios

- Qual a saída do programa abaixo?

```
x = 1.0
y = 2.0
z = 3.0


x = -x
y = y - 1
z = z + x
z = z + x - y
print("x =", x, ", y =", y, ", z =", z)
```

Exercícios

1. Faça um programa que leia o nome, a idade, a altura, o peso e a nacionalidade do usuário e escreva essas informações na forma de um parágrafo de apresentação
2. Faça um programa que exiba o perímetro de uma circunferência a partir do seu raio
3. Faça um programa que leia dois pontos num espaço bidimensional e calcule a distância entre esses pontos

Exercícios

4. Faça um programa que informe a distância em quilômetros de um raio para o observador
   - O observador deve informar o tempo (em segundos) transcorrido entre ver o raio e ouvir o trovão
   - Assuma que a velocidade do som seja 340 m/s

Referências

- Slides preparados pelos professores Vanessa Braganholo, Aline Paes e Leonardo Murta.

Organização de programas em Python

Raquel Bravo
raquel@ic.uff.br